## IN THE CLAIMS

**Please amend the claims as follows:**

Claim 1 (Currently Amended): A tamper resistant microprocessor that executes a plurality of programs in parallel under a multi-task programming environment, comprising:

a decryption unit configured to read out an execution code or data of one of a plurality of encrypted programs and decrypt the execution code or data by using a prescribed encryption key corresponding to the read-out encrypted program, according to a decryption request from a cache memory control unit;

a cache memory configured to store the execution code or data decrypted by the decryption unit into one of cache lines provided in the cache memory, each cache line having a secret protection attribute holding section for storing an actual encryption key used in decrypting the execution code or data, the execution code or data stored in the cache memory remaining even after each program terminates; and

the cache memory control unit configured to process a reading request for the execution code or data to be acquired from the decryption unit or the cache memory such that, if the execution code or data exists in the cache memory and the actual encryption key stored in the secret protection attribute holding section of a cache line that stores the existent execution code or data is identical with the prescribed key corresponding to a program that issues the reading request, the execution code or data in the cache memory is read out, and if the execution code or data does not exist in the cache memory or the actual encryption key is not identical with the prescribed key, the execution code or data is read out from an external memory device.

Claim 2 (Previously Presented): The tamper resistant microprocessor of claim 1, further comprising:

a key value register configured to store the prescribed encryption key, which is updated at an occasion of executing each encrypted program;

wherein the cache memory control unit judges whether the actual encryption key stored in the secret protection attribute holding section of a cache line that stores the existent execution code or data is identical with the prescribed key stored in the key value register.

Claim 3 (Currently Amended): The tamper resistant microprocessor of claim *2,* wherein the cache memory stores data decrypted by the decryption unit, and the cache memory control unit writes a processing result of the data into the cache memory, while storing the prescribed encryption key stored in the key value register into the secret protection attribute holding section of a cache line [tine] for the data.

Claim 4 (Currently Amended): The tamper resistant microprocessor of claim 1, wherein the cache memory stores data decrypted by the decryption unit, and the cache memory control unit encrypts a processing result of the data by using the actual encryption key stored in the secret protection attribute holding section of a cache line for the data, and writes encrypted data into [[an]] the external memory device.

Claim 5 (Currently Amended): A data access control method by a cache memory implemented processor that executes a plurality of programs in parallel under a multi-task programming environment, comprising:

reading out an execution code or data one of a plurality of encrypted programs and decrypting the execution code or data by using a prescribed encryption key corresponding to the read-out encrypted program, according to a decryption request;

storing the execution code or data decrypted by the reading and decrypting step, into one of cache lines provided in a cache memory, each cache line having a secret protection attribute holding section for storing an actual encryption key used in decrypting the execution code or data, the execution code or data stored in the cache memory remaining even after each program terminates; and

processing a reading request for the execution code or data to be acquired from a decryption unit or the cache memory such that, if the execution code or data exists in the cache memory and the actual encryption key stored in the secret protection attribute holding section of a cache line that stores the existent execution code or data is identical with the prescribed key corresponding to a program that issues the reading request, the execution code or data in the cache memory is read out, and if the execution code or data does not exist in the cache memory or the actual encryption key is not identical with the prescribed key, the execution code or data is read out from an external memory device.

Claim 6 (Previously Presented): The data access control method of claim 5, further comprising:

storing the prescribed encryption key, which is updated at an occasion of executing each encrypted program, into a key value register;

wherein whether the actual encryption key stored in the secret protection attribute holding section of a cache line that stores the existent execution code or data is identical with the prescribed key stored in the key value register is judged.

Claim 7 (Previously Presented): The data access control method of claim 6, wherein the cache memory stores data decrypted by the reading and decrypting step, and the data access control method further comprises writing a processing result of the data into the cache

memory, while storing the prescribed encryption key stored in the key value register into the secret protection attribute holding section of a cache line for the data.

Claim 8 (Currently Amended): The data access control method of claim 5, wherein the cache memory stores data decrypted by the reading and decrypting step, and the data access control method further comprises encrypting a processing result of the data by using the actual encryption key stored in the secret protection attribute holding section of a cache line for the data, and writing encrypted data into [[an]] the external memory device.